```java
package test1;

//Scanner is a class for input
import java.util.Scanner;//#include
<string.h>


//Object Oriented Programming ->
Working with OBJECTS

//Class -> Blueprint for objects
//We dont work with CLASSES
//We work with Objects
//And since objects are born from
class
//class =blueprint for how the objects
created from them will behave


//Encapsulation -> See bottom
//Inheritence
//Polymorphism

//Dont have to read this yet//
////////////////////////////////////////
abstract class LivingThing{
```

```
    abstract void speak(); //Any class that
inherit this class must create their
own speak();
}

class Human extends LivingThing{

    //Every objects born from Human
will have these properties
    int age;
    String name;
    float height;
    int password;

    void speak() { //Create our own version
of speak();
        System.out.println("JoJo");
    }

}

class Animal extends LivingThing{
    void speak() {

    }
}

class Car{
    int plate;
```

```
    String name;
}

public class AAA{

    //main (is the same as int main() in C)
    public static void main(String args[]) {

        //Class obj_name =new Class();

        Human Lek=new Human(); //Lek is a
Human
        Human Ptr=new Human(); //Ptr is a
Human

        Car tesla=new Car();

        Lek.age=4;   //Lek age doesn't
have to care for Ptr age
        Ptr.age=1000;//They are not
connected at all
                     //They are objects
of type Human
                     //Objects from a
same class do not have to care for
each other
```

```
                    //Can assign,
destroy, an object without affecting
another object from the same class
        int x=12;

        Scanner barcode=new
Scanner(System.in);
        String name=barcode.nextLine();
        System.out.println("Name is "+
name);


        System.out.println("Lek ="+Lek.age+
"Ptr ="+Ptr.age);
        System.out.println("X is "+x);
        System.out.println("X is "+
Lek.name);
        System.out.println("X is "+(Lek.age+
Ptr.age));

    }
////////////////////////////////////////////


    //Notes
    //
    // Human                    class
```

```
    //  /   \
    //Lek  Somphong                objects
    //
    //Somphone and Lek doesn't have the
same age
    //But both Somphone and Lek have
the ability to have "age", since they
came from Human class

    //Encapsulation (Data must be
accessed in a specify way)
    //   SET    ->[ PASSWORD ]-> GET
    //Data is protected

    //Inheritence(Child class RECEIVE
every things from the parent class)
    // TREE        ->    Grass
    //-Height          -Height
    //   -Name         -Name
    //      -Weight (If a child is a
musician, his father doesn't have to
be a musician)

    //   Polymorphism(One declaration,
multiple definitions)
```

```
    //  Animal ->  speak (any class that
inherited me must create their own
version of speak);
    //  Dog    -> (wood)
    //  Cat    -> (Meow)
    //  Fish   -> (Pog)


}
```